Assignment Report

Hybrid Algorithm

Sayed Kotb Sayed

29th September, 2018

The Algorithm

I have chosen to implement a hybrid algorithm that uses both Merge Sort and Insertion Sort to compose an algorithm that runs faster than both of the formerly mentioned algorithms.

The paradigm is that we know that Insertion Sort is far more efficient on small datasets, and Merge Sort provides an efficient linear method to combine sorted small arrays into sorted larger arrays.

So the algorithm starts by dividing the dataset into small parts of size 32 elements each, and then runs Insertion Sort on each of these small parts to sort them. Then the algorithm proceeds by merging these sorted parts. The algorithm merges parts of sizes 32 elements, and then merges parts of sizes 64 elements and then 128 elements, etc. The algorithm is proved to be perform better when the size increments are powers of 2.

The algorithm performs far better than a conventional Merge Sort, and it's competitive against Quick Sort on unsorted data (loses by a small margin) but beats Quick Sort on sorted data by a large margin. The run time is not dependent on the nature of data, so it is data independent.

The algorithm is called TimSort which is used in the built-in functions in both python and Java.

Time Complexity (Worst case): N.log(N) but faster than a conventional Merge Sort.

Space Complexity: O(N) for the merging part of the algorithm.